 # ADORDD
****

ADORDD for (x)Harbour its ready!

PLEASE MAKE A DONATION!

We will all profit for sure from such development so I think it's fair to ask everyone to contribute with a minimal importance of 20 Euros (because of the PayPal costs) for a good cause.

Go to http://ajusera.com/ scroll down and click PayPal button "Doar" They can send you a contribution receipt if you send them a email with your details.

This is a non-profit organization that performs an important social work supporting the elderly, young problematic, food distribution to the needy, etc. This organization is based on the work of Father Jeronimo Usera (Spanish) well known in Spain, Portugal and South America.

You can get more information at ajusera.com (sorry its only in portuguese)

Steps to start working with **ADORDD**:

==========

Step 1: Just add adordd.prg to your project and include adordd.ch

Step 2: Set parameters as in tryadordd.prg (see in adordd.ch for syntax) and add it in your main.prg like this:

```
// Load ADORDD
REQUEST ADORDD, ADOVERSION, RECSIZE
RddRegister( "ADORDD",1 )
RddSetDefault( "ADORDD" )
```

A) SET ADO TABLES INDEX LIST TO ….

This Set is used by the SQL engine to build select with order by.

Thus the fields must be separated by comma and it can include SQL functions or ASC DESC This Set cannot include Clipper/(x)Harbour functions as they are unknown to SQL.

Example:

```
SET ADO TABLES INDEX LIST TO {
{"TABLE1",{"FIRST","FIRST DESC"} }, {"TABLE2"
,{"CODID","CODID"}} }
```

B) SET ADODBF TABLES INDEX LIST TO…

This Set is used to evaluate Clipper/(x)Harbour expressions such as:

```
&( indexkey( 0 ) )

OrdKey( )

Etc.
```

So it must contain your actual real index expressions.

Example:

```
SET ADODBF TABLES INDEX LIST TO {
{"TABLE1",{"FIRST","FIRST"} }, {"TABLE2"
,{"CODID","STR(CODID,2,0)"}} }
```

C) SET ADO TEMPORAY NAMES INDEX LIST TO…

Indicates the names used for temporary files at SQL level.

It must start by TMP or TEMP but can be "TMPROGER"

These temporary files are mainly used for temporary indexes created in the SQL server as TEMPORARY and automatically destroyed after connection ends.

They are only visible to the user that created them.

Example:

```
SET ADO TEMPORAY NAMES INDEX LIST TO
{"TMP","TEMP"}
```

D) SET ADO FIELDRECNO TABLES LIST TO …

This Set lets you indicate a different autoinc field of the default per table to be used as recno().

Example:

```
SET ADO FIELDRECNO TABLES LIST TO
{{"TABLE1","HBRECNO"},{"TABLE2","HBRECNO"}}
```

E) SET ADO DEFAULT RECNO FIELD TO…

This Set indicates the default field name to be used as recno in all tables besides the mentioned above.

Example:

```
SET ADO DEFAULT RECNO FIELD TO "HBRECNO"
```

ATTENTION:

The D and or E sets are absolutely necessary and without them the navigation with ADORDD might be unpredictable.

F) SET ADO DEFAULT DATABASE TO … SERVER TO … ENGINE TO … USER TO … PASSWORD TO …

This Set indicates the default server and database and authentication parameters we are using.

Connection gets established here.

The engines supported by ADORDD are ACCESS, MYSQL, ORACLE, INFORMIX, MSSQL, FIREBIRD, POSTGRE, ANYWHERE, DBASE, SQLITE, FOXPRO , ADS

Example:

```
// Access
SET ADO DEFAULT DATABASE TO
"D:\LUCAS\TEST2.mdb"
SERVER TO "" ENGINE TO ACCESS USER TO ""
PASSWORD TO  ""

// MySQL
SET ADO DEFAULT DATABASE TO cDataBase SERVER
TO cServer ENGINE TO MYSQL USER TO cUser
PASSWORD TO cPassWord
```

G) SET ADO LOCK CONTROL SHAREPATH TO … RDD TO …

This set enables ADORDD to assure locking records and exclusive use of files as any other RDD.

By default it's not necessary to activate it once the SQL engine take care of locking. Thus you can use:

```
SET ADO FORCE LOCK OFF
```

You need to supply a path where ADORDD creates the tlocks.dbf file to control this.

This RDD file must be a RDD working with locks such as DBFCDX.

This is not a SQL table and if you need to work in WAN and need lock control you will need to:

a) The connection to SQL server

b) Ex a VPN where you can access this share.

Example:

```
SET ADO LOCK CONTROL SHAREPATH TO "D:\LUCAS"
RDD TO "DBFCDX"
```

Step 3: You can start working with ADORDD like a DBFCDX

No code change in your applications with the exceptions of:

Index expressions the variables must be evaluated before sending it to ADORDD.

Deleted records are immediately out of the table and cannot be recovered again. Thus any operations on deleted records must occur before delete the record or an error will occur.

Operations like:

```
delete all

while...

    if lconditon

        recall record
```

must be inverted to:

```
while....

    if !lcondtion
```

```
            delete record
```

and thats it!

Now if you need to upload your tables to any SQL you can do it with:

```
use "table" via "dbfcdx"

copy "table" to "sqtable" via "adordd

use "sqltable"

//you can use a table in a new connection

Use "ctable@connection string" alias
"whatever"
```

<u>Funtions you can call in adordd.prg to use in our application :</u>

```
ADOVERSION() Returns adordd version

ADOSEEKSQL( nWA, lSoftSeek, cKey, lFindLast )
//returns a set of records meeting seek key

ADOBEGINTRANS(nWa)

ADOCOMMITTRANS(nWa)

ADOROLLBACKTRANS(nWa)

ADORESETSEEK( nWa ) //resets the recordset to
previous before call ADOSEEKSQL()

hb_adoRddGetConnection( nWorkArea ) Returns
the connection for the workarea

hb_adoRddGetRecordSet( nWorkArea ) Returns the
recordset for the nWorkArea
```

hb_adoRddGetTableName(nWorkArea) Returns table name for the nWorkArea

hb_adoRddExistsTable(oCon,cTable, cIndex) Returns .t. if table or table and or index exist on the DB

hb_adoRddDrop(oCon, cTable, cIndex, DBEngine) Drops (delete) table or index in the DB

hb_GetAdoConnection() Returns ado default connection

Revision English manual: 1.0; Antonio H Ferreira; 17.06.2015